

TI Designs: TIDEP-0089

用于 Sitara™ AM335x 的 CC-Link IE Field Basic 主站和从站参考设计



说明

此 CC-Link IE Field Basic 参考设计基于 Sitara™AM335x 处理器，并同时采用处理器 SDK RTOS 和处理器 SDK Linux®。对于 RTOS，此设计使用了一个网络开发套件 (NDK) 传输层，并且在 NIMU (EMAC) 和 NIMU_ICSS (PRU-ICSS 双 EMAC 固件) 层上的示例都支持 RTOS。对于 Linux，此设计使用可基于 EMAC 或 PRU-ICSS 的 Linux 网络栈。该实施方案使用主站或从站配置。

资源

TIDEP-0089	设计文件夹
AM3359	产品文件夹
DP83822I	产品文件夹
TMDSICE3359	产品文件夹



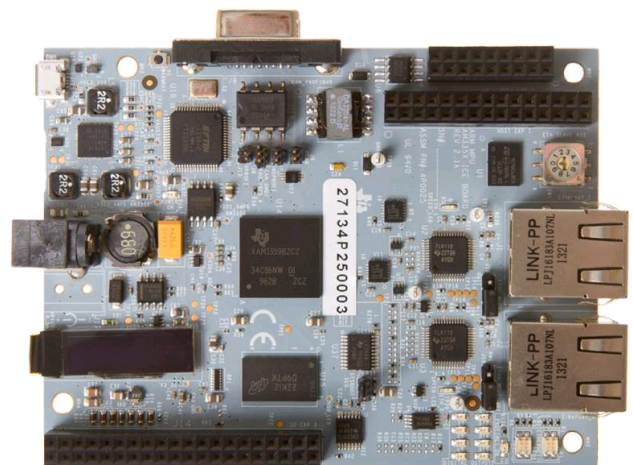
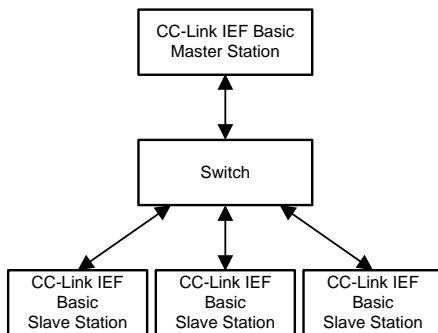
咨询我们的 E2E 专家

特性

- CC-Link 工业以太网 Field Basic 主站和从站参考设计
- 支持 100Mbps
- 符合无缝消息协议 (SLMP) 标准 - 从站
- 最多支持 64 个从站 - 主站
- 每个组占用的最多站数为 16
- 可通过与处理器 SDK 一起打包的源代码进行完全自定义
- 也可通过处理器 SDK 在其他 EVM 上获得支持

应用

- 工业以太网
- 伺服驱动器和运动控制
- 可编程逻辑控制器 (PLC)
- 工业通信模块
- 工业输入/输出 (IO) 模块
- 工业传感器和传动器



该 TI 参考设计末尾的重要声明表述了授权使用、知识产权问题和其他重要的免责声明和信息。

1 System Description

Control and Communication Link (CC-Link) is an open network administered as a fully-open architecture by the CC-Link Partner Association (CLPA). CC-Link guarantees 10-Mbit/s performance across the fieldbus network, regardless of device type, which eliminates hidden bottle necks that are common with other open systems. For the Industrial Ethernet version of CC-Link IE Field, the speed is 1 Gbit. CC-Link offers the freedom to integrate a wide variety of automation components into a single, seamless automation system on the network. CC-Link is available in multiple formats: CC-Link, CC-Link Safety, CC-Link IE (Industrial Ethernet) Control, and CC-Link IE Field.

CC-Link IE Field Basic (or *IEF Basic*) is a new addition to the family of CC-Link IE open network technologies that enable device vendors to easily add CC-Link IE compatibility to any product with a 100-Mbit Ethernet port. IEF Basic is easily implemented on devices or master controllers by software alone, which enables added compatibility to existing products without any hardware modification. IEF Basic's stack is compatible with TCP/IP and UDP/IP; the stack blends seamlessly with other Ethernet technologies (including switches, cables, connectors, and wireless systems). Finally, a master controller for the network is also purely software based, so any industrial PC or other Ethernet equipped controller can be rapidly deployed to run an IEF Basic network without requiring any special interface cards, driver development, or other additional work. The devices all communicate using cyclic (synchronous) exchange of data, which means network updates are performed on a regular, deterministic schedule.

Seamless Message Protocol (SLMP) is a common protocol for achieving seamless communication between applications without awareness of network hierarchy or boundaries between the CC-Link family network and general-purpose Ethernet devices. SLMP is implemented on network hierarchies, such as TCP/IP, CC-Link IE, and CC-Link. SLMP implementation makes client and server-type and push-type communication possible between general-purpose Ethernet devices, CC-Link IE devices, and CC-Link devices.[\[21\]](#)

2 System Overview

This section gives a basic overview of CC-Link IEF Basic protocol. Most of this information here is from the IEF Basic User Guide. For additional details on IEF Basic, refer to the CC-Link IEF Basic User Guide available from [CLPA](#).

2.1 Block Diagram

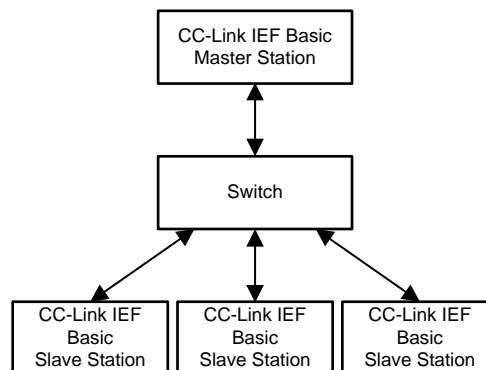


图 1. CC-Link Block Diagram

2.2 Highlighted Products

2.2.1 AM3359

- Up to 1-GHz Sitara ARM® Cortex™-A8 32-bit RISC processor
- NEON™ SIMD coprocessor
- 32KB of L1 Instruction and 32KB of data cache with single-error detection (parity)
- 256KB of L2 cache with error correcting code (ECC)
- 176KB of on-chip boot ROM
- 64KB of dedicated RAM
- Emulation and debug - JTAG
- Interrupt controller (up to 128 interrupt requests)
- PRU-ICSS:
 - Supports protocols such as EtherCAT®, PROFIBUS®, PROFINET®, EtherNet/IP™, and more
 - Two PRUs 32-bit load and store RISC processor capable of running at 200 MHz
 - 8KB of instruction RAM with single-error detection (parity)
 - 8KB of data RAM with single-error detection (parity)
 - Single-cycle, 32-bit multiplier with 64-bit accumulator
 - Enhanced GPIO module provides shift-in or shift-out support and parallel latch on external signal
 - 12KB of shared RAM with single-error detection (parity)
 - Three 120-byte register banks accessible by each PRU INTC for handling system input events
 - Local interconnect bus for connecting internal and external masters to the resources inside the PRU-ICSS
 - Peripherals inside the PRU-ICSS:
 - One universal asynchronous receiver and transmitter (UART) port with flow control pins that supports up to 12 Mbps
 - One enhanced capture (eCAP) module
 - Two MII Ethernet ports that support industrial Ethernet, such as EtherCAT
 - One management data input and output (MDIO) port
 - On-chip memory (shared L3 RAM):
 - 64KB of general-purpose on-chip memory controller (OCMC) RAM
 - Accessible to all masters
- External memory interfaces (EMIF):
 - mDDR(LPDDR), DDR2, DDR3, and DDR3L controller:
 - mDDR: 200-MHz clock (400-MHz data rate)
 - DDR2: 266-MHz clock (532-MHz data rate)
 - DDR3: 400-MHz clock (800-MHz data rate)
 - DDR3L: 400-MHz clock (800-MHz data rate)
 - 16-bit data bus
 - 1GB of total addressable space

- Supports one x16 or two x8 memory device configurations
- General-purpose memory controller (GPMC)
- Flexible 8-bit and 16-bit asynchronous memory interface with up to seven chip selects (NAND, OR, Muxed-NOR, or SRAM)
- Uses BCH code to support 4-, 8-, or 16-bit ECC
- Uses hamming code to support 1-bit ECC

See the *AM335x Sitara Processors*[1] datasheet for a complete list of features.

2.2.2 DP83822I

- IEEE 802.3u compliant: 100BASE-FX, 100BASETX and 10BASE-Te
- MII, RMII, and RGMII MAC Interfaces
- Low-power single supply options:
 - 1.8-V average (AVD) < 120 mW
 - 3.3-V AVD < 220 mW
- ±16-kV HBM ESD Protection
- ±8-kV IEC 61000-4-2 ESD Protection
- Start of frame detect for IEEE 1588 time stamp
- Fast link-down timing
- Auto-crossover in force modes
- Operating temperature: –40°C to 125°C
- IO voltages: 3.3 V, 2.5 V, and 1.8 V
- Power savings features:
 - Energy efficient Ethernet (EEE) IEEE 802.3az
 - Wake-on-LAN (WoL) support with magic packet detection
 - Programmable energy savings modes
- Cable diagnostics
- BIST
- Management data clock (MDC) and MDIO interface

See the *DP83822 Robust, Low Power 10/100 Mbps Ethernet Physical Layer Transceiver*[2] datasheet for a complete list of features.

2.2.3 TMSICE3359 ICE EVM

Hardware specifications:

- AM3359 ARM Cortex-A8
- DDR3, NOR flash, and SPI flash
- Organize light-emitting diode (OLED) display
- TPS65910 power management 24-V power supply
- USB cable for JTAG interface and serial console

PRU-ICSS subsystem for industrial communication, capable of supporting:

- CC-Link IEF Basic Master/Slave
- PROFIBUS interface
- CANOpen
- EtherNet/IP
- PROFINET
- Sercos III
- Digital IO
- SPI
- UART
- JTAG

2.3 System Design Theory

2.3.1 CC-Link

The following is an overview of the characteristics of IEF Basic:

1. Realization of cyclic transmission using IP packets
 - Using an Internet Protocol with an EtherType of Ethernet frame, IP packets allow the realization of cyclic transmission for periodically updating linked devices.
 - Protocols using other IP packets (including HTTP, FTP, SLMP, and so on) can transmit on the same IP network.
 - Data periodically communicates between the master station and slave stations using link devices.
2. Defining protocol at the application layer
 - Because the application layer defines the protocol, there is no required special hardware to realize IEF Basic, and implementing the software allows cyclic transmission realization.
3. Simple protocols
 - Protocol is request-response type with a simple status and status transition that is managed at the station. In addition, the small number of frame types allows simple implementation in machines.
4. Inheritance of CC-Link IE Field Network protocols
 - Because the primary components in CC-Link IE Field are inherited as much as necessary, configuration of the IEF Basic network is similar to that of the CC-Link IE Field Network.

2.3.1.1 Types of Communication

CC-Link IEF Basic performs transmission and reception of frames related to cyclic transmission. By storing the information related with cyclic transmission and station information within this frame type, a single frame can be used for cyclic transmission and network management functions.

2.3.1.2 Protocol Hierarchy

图 2 shows the protocol hierarchy of CC-Link IEF Basic.

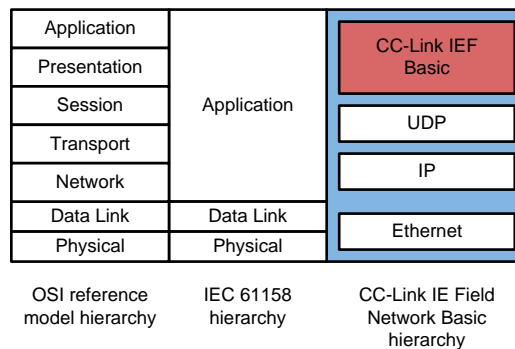


图 2. CC-Link IEF Basic Overview

2.3.2 SLMP

图 3 shows an overview of SLMP.

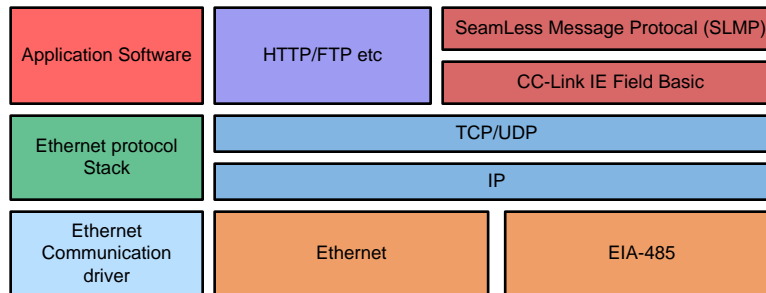


图 3. SLMP Overview

2.3.2.1 Features

SLMP offers the following features:

1. Access to network information
 - SLMP communication makes it possible to access (read and write) information (stored memory) within a server from a client. This stored memory may include internal memory, drive memory, expanded module memory, and so on as well as other information, such as device operation status information, production status information, and sequence program and parameter files.
2. Control from a remote location
 - SLMP-based communication makes it possible to perform server remote control from a client. The control operations include remote control (remote run, stop, pause, clear latch, reset), remote password setup and clearance, and error code initialization.
3. On-demand communication
 - SLMP-based communication makes it possible to transmit urgent data without request from the server to a client, which is called on-demand communication.
4. Efficient data collection
 - Using SLMP, the client can collect data within the server. If the data to be collected is registered in the service in advance, the data distributes without a request by the client.
5. Access to device information
 - SLMP provides a meaning of directly accessing device information. For example, the connected device is automatically detected using the SLMP command and parameter setting. Monitoring and diagnosis can be performed for any device using the same procedure.
6. Integration of other open networks
 - For transient transmission in other open networks, access is enabled from CC-Link Family Network to other open networks from the conversion model using SLMP. For example, the connected devices in other open networks are automatically detected using SLMP, and parameter setting and diagnosis can be performed for any device using the same procedure.

2.3.3 Protocol Overview

The following sections show the sequencing of the communication between master and slave station in an IEF Basic network.[20]

2.3.3.1 Overall Processing Sequence

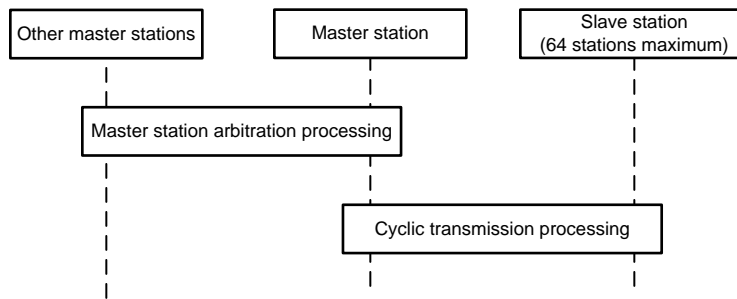


图 4. Basic Sequence of Processing

The processing is performed in the following sequence:

1. The master station performs master station arbitration processing.
2. If master station duplication was not detected in master station arbitration processing, the master station performs cyclic transmission processing.

2.3.3.2 Master Station Arbitration

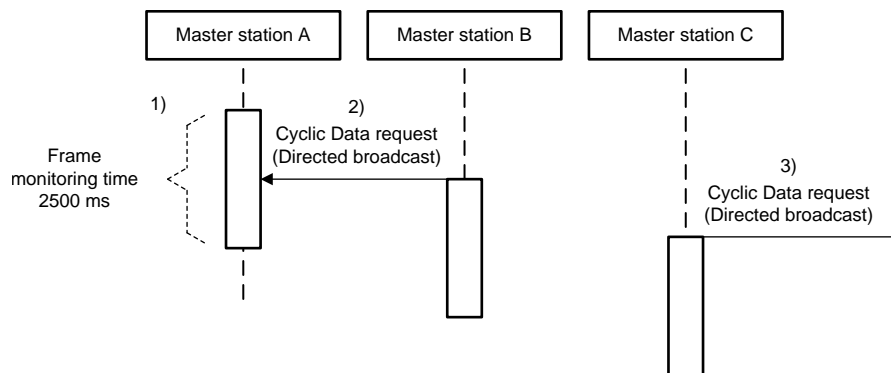


图 5. Master Station Arbitration

The processing is performed in the following sequence:

1. The master station monitors the frame for 2500 ms to check whether the station receives *Cyclic Data* command requests from other master stations, as master station arbitration, prior to performing cyclic transmission processing with the slave station.
2. If the master station receives a Cyclic Data command request, the station judges that there is master station duplication.
3. If the master station does not receive a Cyclic Data command request, the station judges that there is no master station duplication.
4. The master station performs cyclic transmission processing when master station duplication is not detected. If detected, the master station does not perform cyclic transmission processing.

2.3.3.3 Cyclic Transmission

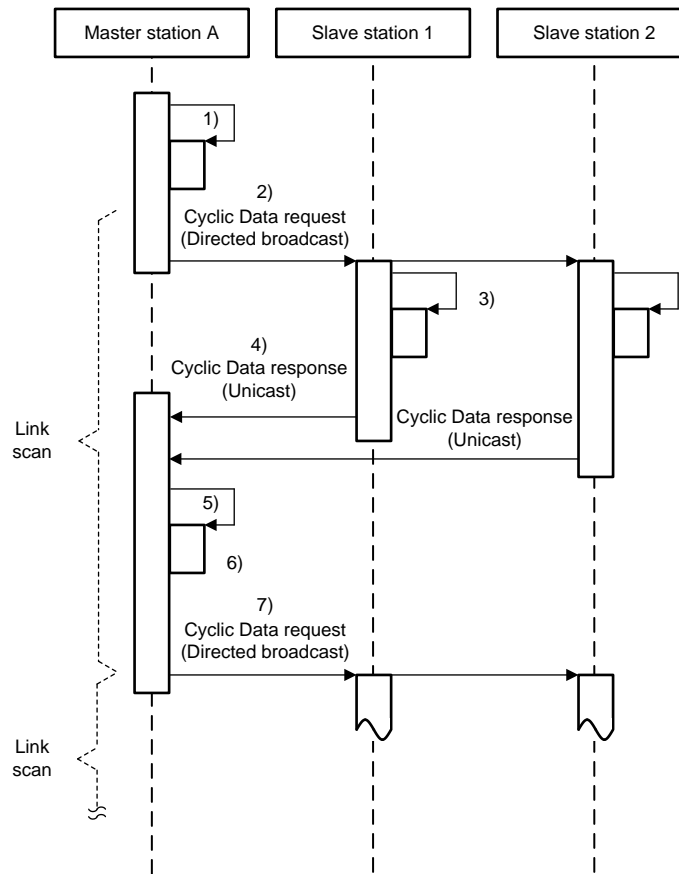


图 6. Cyclic Transmission

The processing is performed in the following sequence:

1. The master station creates cyclic data RY (Remote IO Request bits) and RWw (Remote Register words) before starting a link scan.
2. The master station sends the Cyclic Data command request using a directed broadcast.
3. After receiving a Cyclic Data command request, each slave station transfers the station's specific cyclic data from the request.
4. Each slave station creates its cyclic RX (Remote IO Response bits) and RWr (Remote Register word) data and sends the Cyclic Data command response through a unicast.
5. The master station receives the response from all slave stations with a cyclic transmission status bit turned on. When the constant link scan is used, the master station waits until the constant link scan time elapses.
6. The master station transfers the cyclic data RWr and RX from the Cyclic Data command response and creates the new cyclic data RY and RWw for the next cyclic transmission.
7. Steps 2 to 6 repeat. When multiple groups exist, after step 1 completes, steps 2 to 6 repeat independently for each group.

The upper limit of the link scan time is the total of the response waiting time and the processing time for completion of link scans, such as device transfers. The slave stations process the cyclic data of the Cyclic Data command requests as valid data when the cyclic transmission status bit of the own station is turned on (cyclic data is acquired).

The slave station does not return any command response if the Cyclic Data command request does not include the slave station ID of the own station. If the master station receives a command response from the slave station with a cyclic transmission status bit turned on, and the frame sequence number corresponds with the value of the request message, the master station processes the cyclic data as valid data (cyclic data is acquired).

2.3.4 State Transition

The general state transition of the master station and slave stations of IEF Basic is shown in the following subsections.

2.3.4.1 State Transition of Master Station

The status of the master station consists of a group status and an individual status of each slave station (sub status).

图 7 shows the state transition of a group status of a master station.

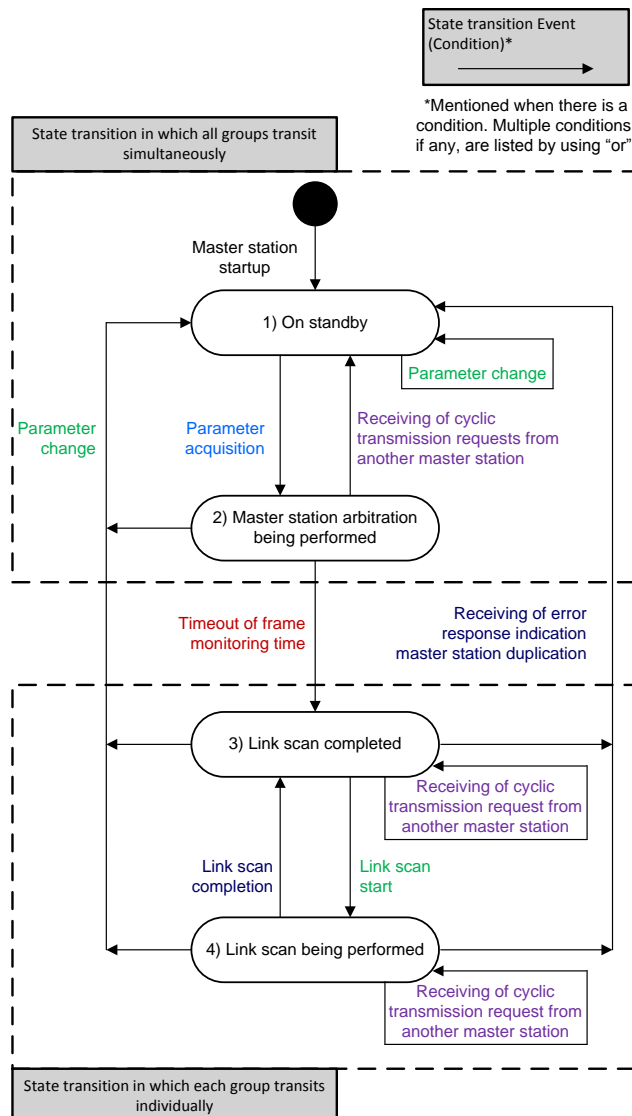


图 7. State Transition of Group Status of Master Station

The group status is the status of each group. When multiple groups exist, there are multiple group statuses and two types of state transition—transition where all groups transit simultaneously and where each group transits individually. Each group status connects with the individual status of each slave station belonging to each group.

图 8 shows the state transition for an individual status of each slave station.

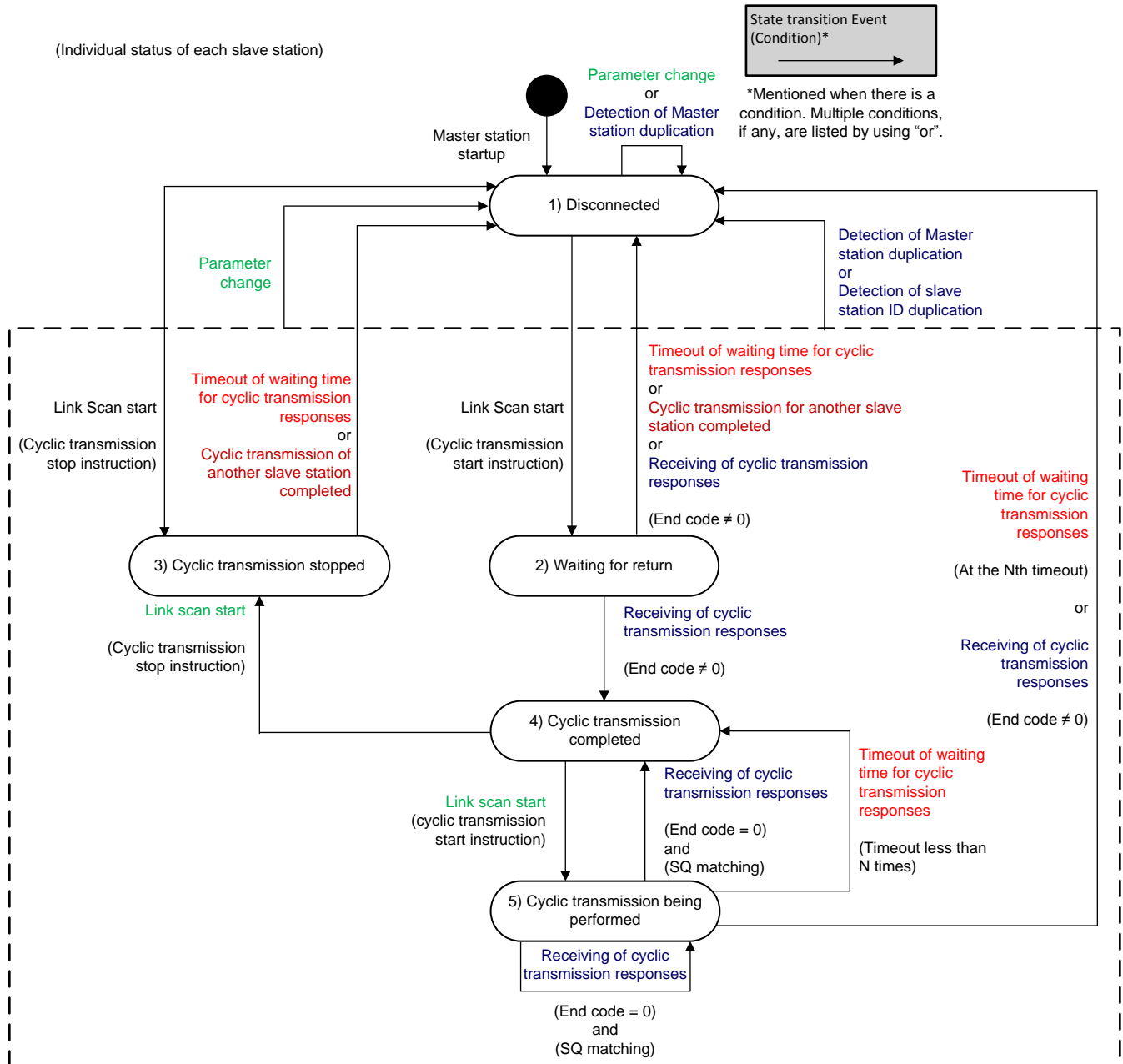


图 8. State Transition Diagram for Individual Status of Each Slave Station Possessed

The master station possesses the individual status of each slave station to control for the number of connected devices. Each status connects with the group status of the group that each slave station belongs.

2.3.4.2 State Transition of Slave Station

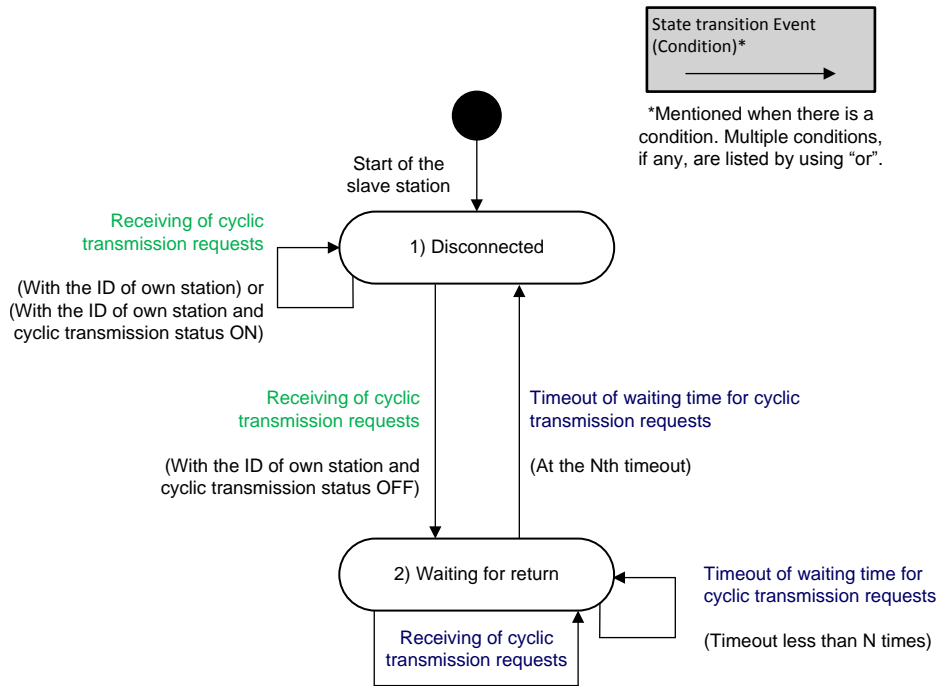


图 9. State Transition Diagram of Slave Station

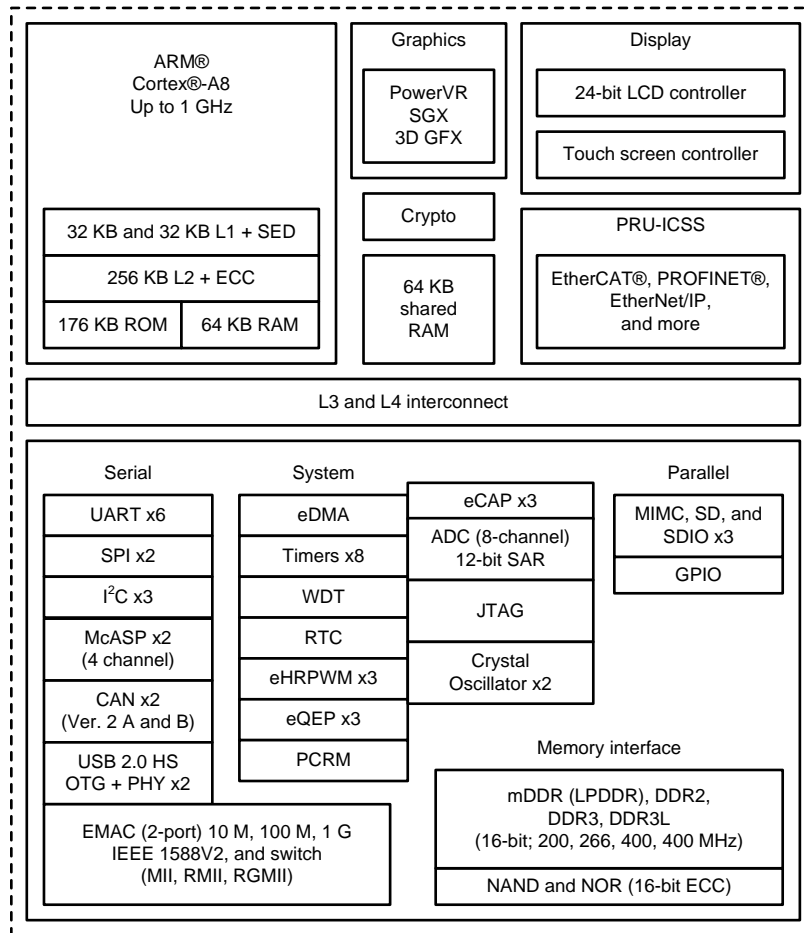
3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

3.1.1 Hardware

This reference design requires the following:

- [AM3359 Sitara Processor](#) (as shown in [图 10](#))



Copyright © 2017, Texas Instruments Incorporated

图 10. Functional Block Diagram of AM335x SOC

- [DP83822I Transceiver](#)

- **TMDSICE3359 ICE EVM** (as shown in 图 11)

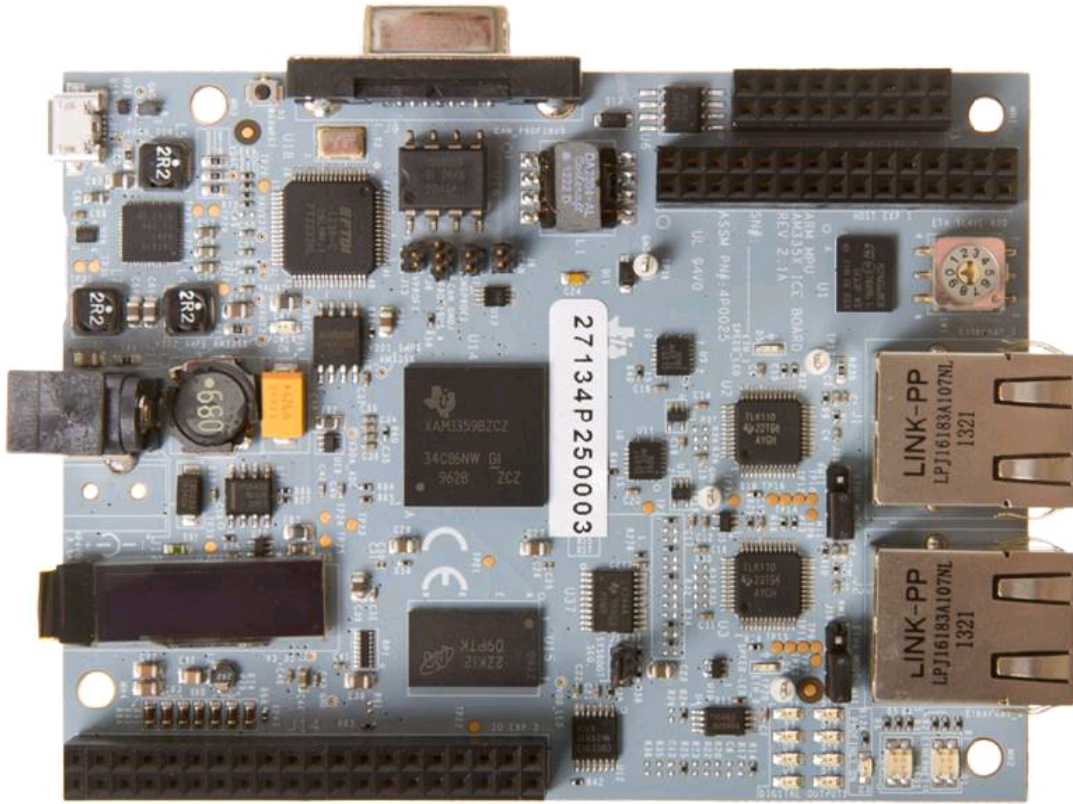


图 11. TMDSICE3359 ICE EVM

3.1.1.1 Additional EVMs Supported

In addition to icev2AM335x board, the IEF Basic master and slave station example also supports other EVMs. 表 1 details the additional supported EVMs.

表 1. Additional EVMs Supported

DEVICE	EVM	IEF Basic on EMAC		IEF Basic on PRU-ICSS	
		Linux	RTOS	Linux	RTOS
AM572x	AM572x EVMxx	X	X		
	AM572x IDK	X	X	X	X
AM571x	AM571x IDK	X	X	X	X
AM437x	AM437x EVM	X	X		
	AM437x SK	X	X		
	AM437x IDK	X	X	X	X
AM335x	AM335x EVM	X	X		
	AM335x BeagleBoneBlack	X	X		
	AM335x SK	X	X		
	AM335x ICE	X	X	X	X
K2G	K2G EVM	X	X		
	K2G ICE EVM	X	X	X	X

3.1.2 Software

3.1.2.1 CC-Link in Processor SDK RTOS

3.1.2.1.1 Software Stack

In *Processor SDK RTOS* the ARM application creates the OS (TI-RTOS) task for supporting various server end functionality.[20] The application creates the network stack for basic networking functionality using *NDK*. The application then initializes the PRU-ICSS subsystem for NIMU_ICSS and CPSW for NIMU.

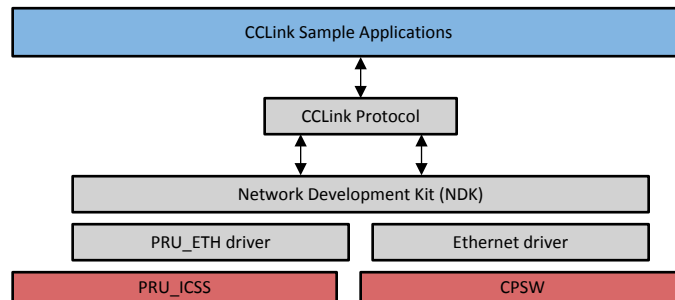


图 12. CC-Link IEF Basic Software Stack

3.1.2.1.1.1 RTOS Adaptation

The general available IEF Basic source code supports Windows® and Linux operating systems. Required modifications enable IEF Basic on RTOS. Most of the modifications are done the on *Hardware Abstraction* layer. The following is the list of changes made. RTOS has a default IP address as *192.168.3.10* for the slave station and *192.168.3.100* for the master station. In order to change this parameter, the RTOS application *.cfg files must be updated.

1. Socket
 - The network layer for RTOS is different than Windows and Linux. The network layer is provided by NDK layer for RTOS. TI NDK is compatible with standard BSD socket layer. All the network functionalities are supported using NDK stack.
2. RTC
 - In case of Windows and Linux, the timing information is extracted from RTC call in both OS. RTOS provides an abstraction layer for RTC call. RTOS configures the timers available in the SOC.
3. SYBBIOS
 - RTOS requires a top-level application to first configure the EVM parameters and set up the board. The application creates the NDK stack and the system configuration. The application then creates a task for IEF Basic application, which runs on top of the stack.
4. UART
 - RTOS provides the output to be printed on UART console.

3.1.2.1.2 Run CC-Link IEF Basic Sample RTOS Application

The following software is required:

- Code Composer Studio™ (CCS) v6 or higher
- PRU Compiler for CCSv6 (install through CCS add-on)
- PROCESSOR SDK RTOS AM335X

Software setup:

1. Install [CCS](#) development tool.
2. Install [PROCESSOR SDK RTOS AM335X](#).
3. Create [application projects](#) depending on target application.
4. Import [IEF Basic application](#) project into CCS. Connect to one of the boards, and run the master application example. Connect to the other board, and run the slave application on it.
5. [Output result](#) will be printed on UART console.

注: Review [6 节](#) for wiki links with additional details.

3.1.2.2 CC-Link in Processor SDK Linux

3.1.2.2.1 Software Stack

In Processor SDK Linux, the IEF Basic application runs on top of the Linux networking, which can be based on either EMAC (CPSW) or PRU-ICSS.

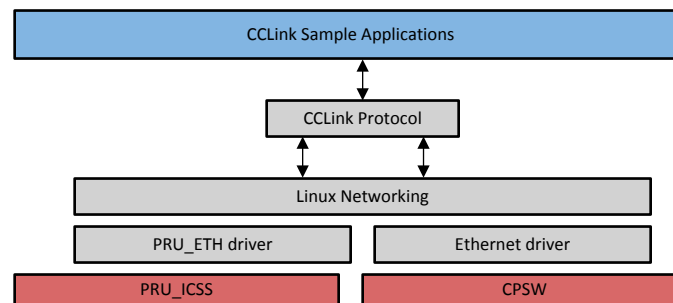


图 13. CC-Link IEF Basic Software Stack

3.1.2.2.2 Run CC-Link IEF Basic Sample Linux Application

The following software is required:

- [PROCESSOR SDK Linux RT AM335X](#)

Software setup:

1. Download [PROCESSOR SDK Linux RT AM335X](#).
2. Follow the instructions on the wiki pages to create SD cards.
 - For a Windows machine, follow the instructions in [Processor SDK Linux Creating a SD Card with Windows](#).
 - For a Linux machine, follow the instructions in [Processor SDK Linux create SD card script](#).
3. Follow the instructions at [Processor SDK Linux CCLINK](#) to obtain the source code of the IEF Basic master and slave sample applications.

To run the IEF Basic [sample application](#):

1. Boot the two icev2AM335x boards with the SD cards inserted.
2. On the master icev2AM335x board, modify *Slave1 IP address* in MasterParameter.csv to use the IP address of the slave icev2AM335x board.
3. Run *Master_sample* application on the master icev2AM335x board.
4. Run *Slave_sample* application on the slave icev2AM335x board.

3.2 Testing and Results

3.2.1 Test Setup

图 14 shows the test setup for the IEF basic master and slave application running on icev2AM335x board.

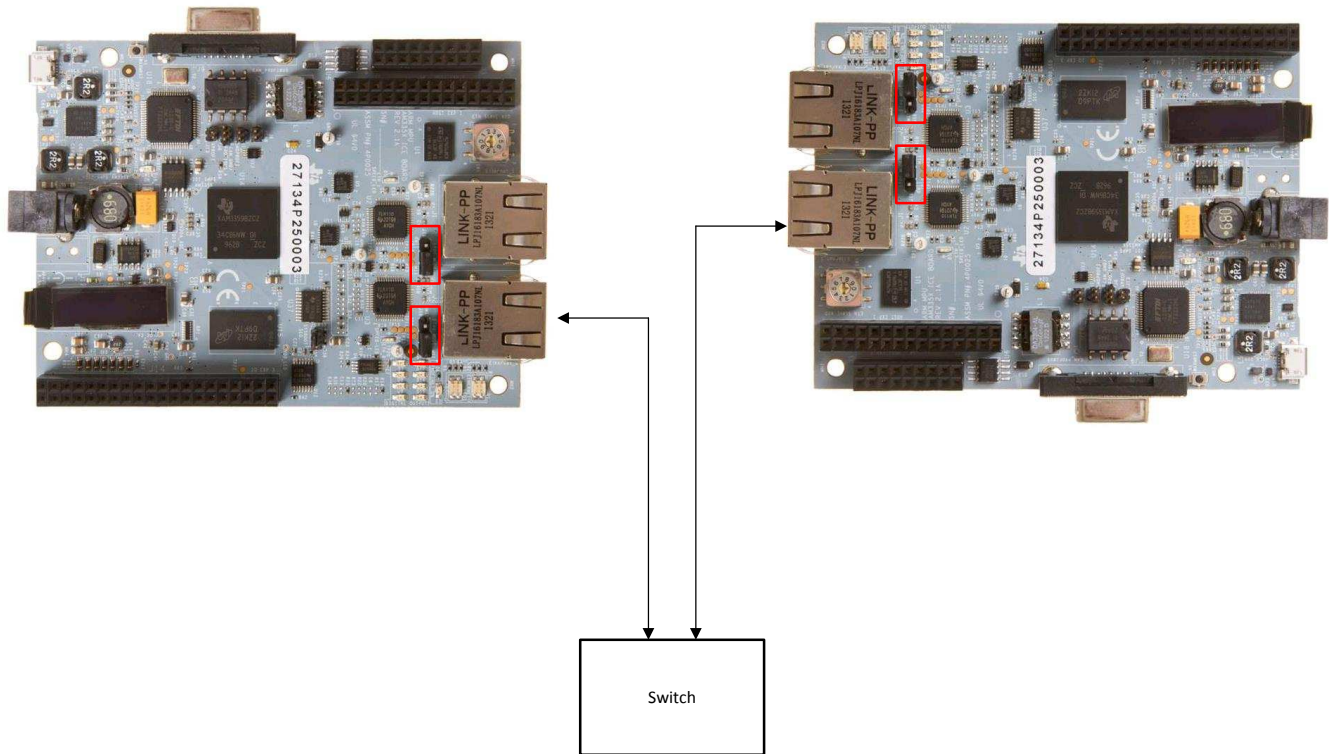


图 14. Hardware Test Setup

Connect port 0 of the design board with an Ethernet cable to a standard switch for both master and slave station. Make sure the jumper setting is correct and based on the type of application demonstrated.

- If running with EMAC, connect the jumper J18 and J19 for both boards into EMAC mode. Hence, connect pin1 and pin2.
- If running with PRU_ICSS, connect the jumper J18 and J19 for both boards into ICSS mode. Hence, connect pin2 and pin3.

In order to get best performance result, do not make any other connection with the switch or hub.

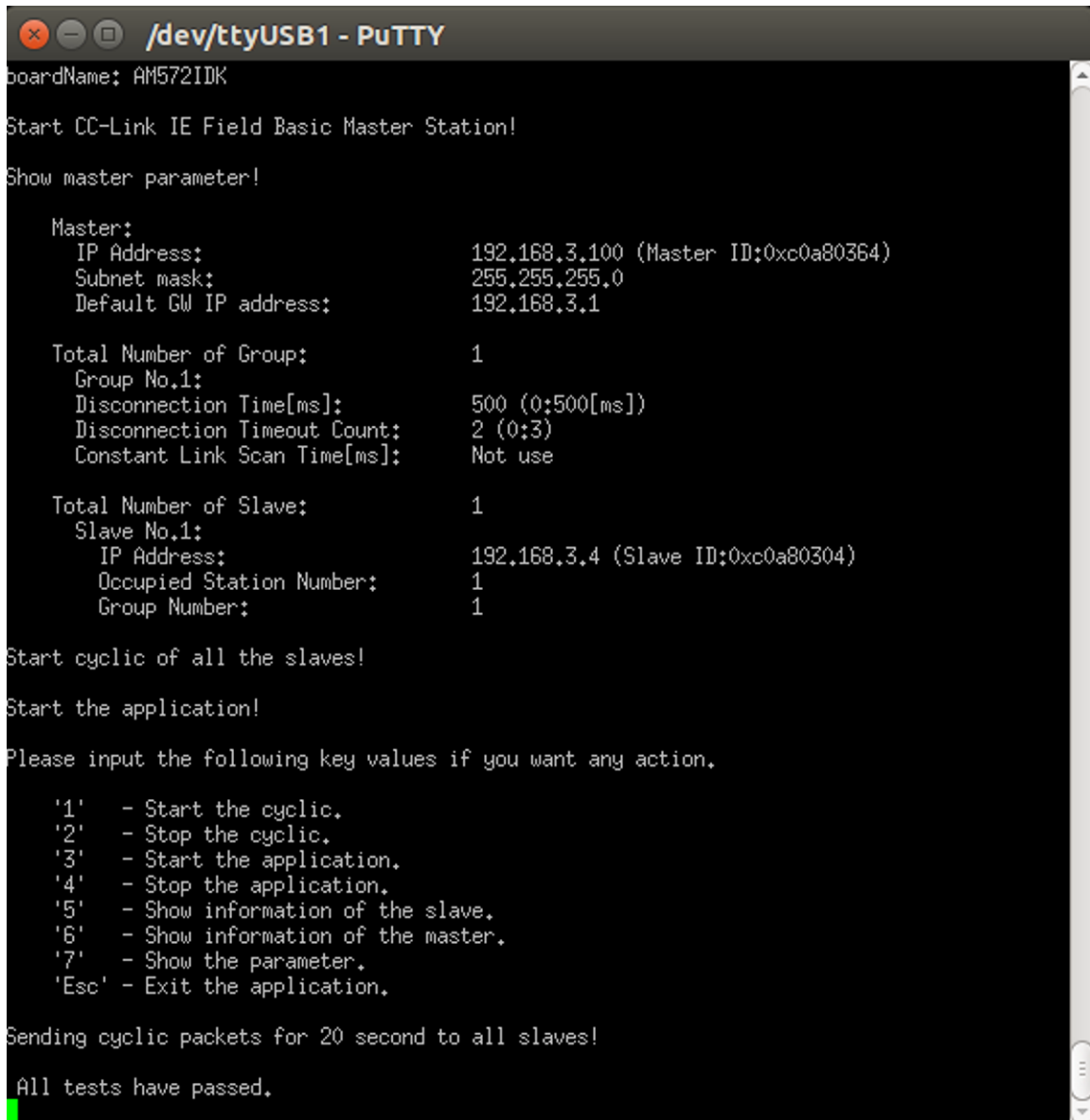
3.2.2 Test Results

3.2.2.1 RTOS

3.2.2.1.1 Sample Output

The following figures show the displays on the UART console when the link is up and communication takes place between slave and master. The default configuration of master and slave would be printed in their respective port.

图 15 shows the master UART console.



```

/dev/ttyUSB1 - PuTTY
boardName: AM572IDK
Start CC-Link IE Field Basic Master Station!
Show master parameter!

Master:
  IP Address:      192.168.3.100 (Master ID:0xc0a80364)
  Subnet mask:    255.255.255.0
  Default GW IP address: 192.168.3.1

Total Number of Group:      1
  Group No.1:
  Disconnection Time[ms]:    500 (0:500[ms])
  Disconnection Timeout Count: 2 (0:3)
  Constant Link Scan Time[ms]: Not use

Total Number of Slave:      1
  Slave No.1:
  IP Address:      192.168.3.4 (Slave ID:0xc0a80304)
  Occupied Station Number: 1
  Group Number:    1

Start cyclic of all the slaves!
Start the application!
Please input the following key values if you want any action.

'1' - Start the cyclic.
'2' - Stop the cyclic.
'3' - Start the application.
'4' - Stop the application.
'5' - Show information of the slave.
'6' - Show information of the master.
'7' - Show the parameter.
'Esc' - Exit the application.

Sending cyclic packets for 20 second to all slaves!
All tests have passed.
    
```

图 15. Master UART Console

图 16 shows the slave UART console.

```

/dev/ttyUSB1 - PuTTY
boardName: AM572IDK

Start CC-Link IE Field Basic Slave Station!

IP Address:          192.168.3.4
Subnet mask:         255.255.255.0
Default GW IP address: 192.168.3.1
Number of Occupied Stations: 1
Cyclic response wait time: 0[ms] (0:Not wait)

Show the cyclic information at 5-second intervals.

Master | Cyclic Data | Link scan time | Time Data

Master          - The control master station.(Master ID / Group Number)
Cyclic Data     - Count of the receive Cyclic Data at intervals.
Link scan time  - The link scan time at intervals,[ms]
Time Data       - The time from the master information data.

CCIEF-BASIC: Master(0xc0a80329/1) | Cyclic Data=9238 | 0.5000[ms] | 2017-06-27 12:11:11
CCIEF-BASIC: Master(0xc0a80329/1) | Cyclic Data=9422 | 0.5000[ms] | 2017-06-27 12:11:16
CCIEF-BASIC: Master(0xc0a80329/1) | Cyclic Data=9346 | 0.5000[ms] | 2017-06-27 12:11:21
CCIEF-BASIC: Master(0xc0a80329/1) | Cyclic Data=9172 | 0.5000[ms] | 2017-06-27 12:11:25
CCIEF-BASIC: Master(0xc0a80329/1) | Cyclic Data=9053 | 0.5000[ms] | 2017-06-27 12:11:30
CCIEF-BASIC: Master(0xc0a80329/1) | Cyclic Data=8916 | 0.5000[ms] | 2017-06-27 12:11:35
CCIEF-BASIC: Master(0xc0a80329/1) | Cyclic Data=9102 | 0.5000[ms] | 2017-06-27 12:11:40

All tests have passed.

```

图 16. Slave UART Console

3.2.2.1.2 Compliance Testing

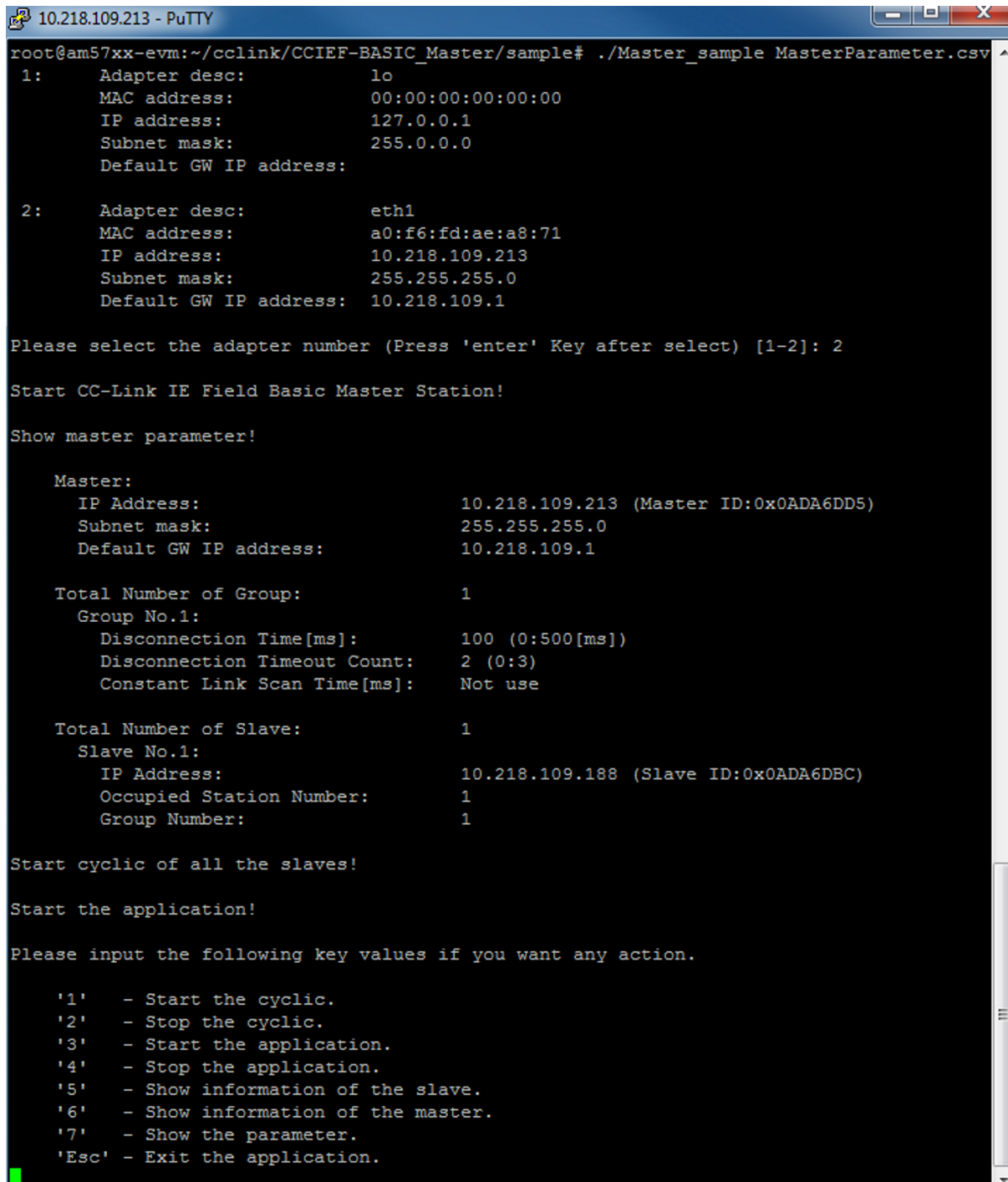
Every IEF Basic application when demonstrated on any platform has to pass the conformance testing for various functionalities of IEF Basic. The conformance test results are sent to CLPA for approval. Upon approval from CLPA, the platform is accepted as CC-Link IEF Basic compliant. See the conformance test results for TI EVMs at [Processor SDK RTOS CCLINK](#).

3.2.2.2 Linux

3.2.2.2.1 Sample Output

The following figures show the displays on the console when communication takes place between the slave and master boards. The default configuration of master and slave would be printed in their respective console.

图 17 shows the master console for the master board.



```

root@am57xx-evm:~/cclink/CCIEF-BASIC_Master/sample# ./Master_sample MasterParameter.csv
1:  Adapter desc:      lo
   MAC address:      00:00:00:00:00:00
   IP address:       127.0.0.1
   Subnet mask:      255.0.0.0
   Default GW IP address:

2:  Adapter desc:      eth1
   MAC address:      a0:f6:fd:ae:a8:71
   IP address:       10.218.109.213
   Subnet mask:      255.255.255.0
   Default GW IP address:  10.218.109.1

Please select the adapter number (Press 'enter' Key after select) [1-2]: 2

Start CC-Link IE Field Basic Master Station!

Show master parameter!

Master:
  IP Address:      10.218.109.213 (Master ID:0x0ADA6DD5)
  Subnet mask:     255.255.255.0
  Default GW IP address:  10.218.109.1

Total Number of Group:      1
  Group No.1:
    Disconnection Time[ms]:  100 (0:500[ms])
    Disconnection Timeout Count:  2 (0:3)
    Constant Link Scan Time[ms]:  Not use

Total Number of Slave:      1
  Slave No.1:
    IP Address:      10.218.109.188 (Slave ID:0x0ADA6DBC)
    Occupied Station Number:  1
    Group Number:     1

Start cyclic of all the slaves!

Start the application!

Please input the following key values if you want any action.

'1'  - Start the cyclic.
'2'  - Stop the cyclic.
'3'  - Start the application.
'4'  - Stop the application.
'5'  - Show information of the slave.
'6'  - Show information of the master.
'7'  - Show the parameter.
'Esc' - Exit the application.
    
```

图 17. Master Console

图 18 shows the slave console for the slave board.

```

10.218.109.188 - PuTTY
root@am57xx-evm:~/cclink/CCIEF-BASIC_Slave/sample# ./Slave_sample SlaveParameter.csv
 1:  Adapter desc:      lo
     MAC address:     00:00:00:00:00:00
     IP address:      127.0.0.1
     Subnet mask:    255.0.0.0
     Default GW IP address:

 2:  Adapter desc:     eth0
     MAC address:     5c:f8:21:3b:f0:bc
     IP address:      10.218.109.188
     Subnet mask:    255.255.255.0
     Default GW IP address: 10.218.109.1

Please select the adapter number (Press 'enter' Key after select) [1-2]: 2

Start CC-Link IE Field Basic Slave Station!

IP Address:      10.218.109.188
Subnet mask:     255.255.255.0
Default GW IP address: 10.218.109.1
Number of Occupied Stations: 16
Cyclic response wait time: 0[ms] (0:Not wait)

Show the cyclic information at 5-second intervals.

Master | Cyclic Data | Link scan time | Time Data

Master          - The control master station. (Master ID / Group Number)
Cyclic Data     - Count of the receive Cyclic Data at intervals.
Link scan time  - The link scan time at intervals. [ms]
Time Data       - The time from the master information data.

CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29117 | 0.172 [ms] | 2017-06-29 04:04:32
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29172 | 0.171 [ms] | 2017-06-29 04:04:37
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29193 | 0.171 [ms] | 2017-06-29 04:04:43
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29079 | 0.172 [ms] | 2017-06-29 04:04:48
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=28795 | 0.174 [ms] | 2017-06-29 04:04:53
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29635 | 0.169 [ms] | 2017-06-29 04:04:58
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29310 | 0.171 [ms] | 2017-06-29 04:05:03
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29904 | 0.167 [ms] | 2017-06-29 04:05:08
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=28673 | 0.174 [ms] | 2017-06-29 04:05:13
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29331 | 0.170 [ms] | 2017-06-29 04:05:18
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=28857 | 0.173 [ms] | 2017-06-29 04:05:24
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=30040 | 0.166 [ms] | 2017-06-29 04:05:29
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29881 | 0.167 [ms] | 2017-06-29 04:05:34
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29072 | 0.172 [ms] | 2017-06-29 04:05:39
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29287 | 0.171 [ms] | 2017-06-29 04:05:44
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=30285 | 0.165 [ms] | 2017-06-29 04:05:49
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29783 | 0.168 [ms] | 2017-06-29 04:05:54
CCIEF-BASIC: Master(0x0ADA6DD5/1) | Cyclic Data=29264 | 0.171 [ms] | 2017-06-29 04:05:59

```

图 18. Slave Console

3.2.2.2.2 Compliance Testing

See the conformance testing results for icev2AM335x with Processor SDK Linux at [Processor SDK Linux CCLINK](#).

3.2.2.3 Additional EVMs

Follow the same procedure as mentioned in [节 3.2.2.1](#) and [节 3.2.2.2](#) for both RTOS and Linux.

3.2.2.3.1 Compliance Testing

3.2.2.3.1.1 RTOS

See the conformance test results for TI EVMs with Processor SDK RTOS at [Processor SDK RTOS CCLINK](#).

3.2.2.3.1.2 Linux

See the conformance test results for TI EVMs with Processor SDK Linux at [Processor SDK Linux CCLINK](#).

4 Design Files

4.1 Schematics

To download the schematics, see the design files at [TIDEP-0089](#).

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDEP-0089](#).

4.3 PCB Layout Recommendations

4.3.1 Layout Prints

To download the layer plots, see the design files at [TIDEP-0089](#).

4.4 Altium Project

To download the Altium project files, see the design files at [TIDEP-0089](#).

4.5 Gerber Files

To download the Gerber files, see the design files at [TIDEP-0089](#).

4.6 Assembly Drawings

To download the assembly drawings, see the design files at [TIDEP-0089](#).

5 Software Files

To download the software files, see the design files at [TIDEP-0089](#).

6 Related Documentation

1. Texas Instruments, [AM335x Sitara Processors](#), Datasheet (SPRS717)
2. Texas Instruments, [DP83822 Robust, Low Power 10/100 Mbps Ethernet Physical Layer Transceiver](#), Datasheet (SNLS505)
3. Texas Instruments, [Processor SDK RTOS IEF Basic](#), Wiki
4. Texas Instruments, [Download CCS](#), Code Composer Studio TI Wiki
5. Texas Instruments, [CCS Getting Started Guide](#), Wiki
6. Texas Instruments, [Processor SDK RTOS_CCS_Setup](#), Wiki
7. Texas Instruments, [Creating and importing examples in Processor SDK RTOS](#), Wiki
8. Texas Instruments, [Processor SDK RTOS AM335X](#), Download Page
9. Texas Instruments, [Processor SDK RTOS AM437X](#), Download Page
10. Texas Instruments, [Processor SDK RTOS K2G](#), Download Page
11. Texas Instruments, [Processor SDK RTOS AM57X](#), Download Page
12. Texas Instruments, [Processor SDK Linux IEF Basic Wiki Page](#), Wiki
13. Texas Instruments, [SDK Linux create SD card script](#), Wiki
14. Texas Instruments, [Processor SDK Linux Creating a SD Card with Windows](#), Wiki
15. Texas Instruments, [Processor SDK Linux RT AM335X](#), Download Page
16. Texas Instruments, [Processor SDK Linux RT AM437X](#), Download Page
17. Texas Instruments, [Processor SDK Linux RT K2G](#), Download Page
18. Texas Instruments, [Processor SDK Linux RT AM57X](#), Download Page
19. Texas Instruments, [Processor SDK Software Page](#), Product Page
20. Texas Instruments, [Category:SYSBIOS](#), Wiki
21. CC-Link Partner Association, [CLPA Reference Material and Support](#)

6.1 商标

Sitara, Code Composer Studio are trademarks of Texas Instruments Incorporated.

Cortex, NEON are trademarks of ARM Limited.

ARM is a registered trademark of ARM Limited.

EtherCAT is a registered trademark of Beckhoff Automation GmbH.

Linux is a registered trademark of Linux Foundation.

Windows is a registered trademark of Microsoft Corporation.

EtherNet/IP is a trademark of ODVA, Inc..

PROFIBUS, PROFINET are registered trademarks of PROFIBUS and PROFINET International (PI).

All other trademarks are the property of their respective owners.

7 Terminology

- CCS - Code Composer Studio
- ICSS - Industrial communication system
- PLC - Programmable logic controller
- PRU - Programmable real-time unit

8 About the Author

SURAJ DAS is a Software Engineer at Texas Instruments, where he is responsible for developing PRU-ICSS based solution for the Catalog segment. Suraj brings to this role his extensive experience in Computer architecture & PRU cores, and has supported Catalog RTOS SDK release for various peripherals. Suraj earned his Master of Engineering degree in Computer Engineering from Virginia Tech in Blacksburg, VA.

有关 TI 设计信息和资源的重要通知

德州仪器 (TI) 公司提供的技术、应用或其他设计建议、服务或信息，包括但不限于与评估模块有关的参考设计和材料（总称“TI 资源”），旨在帮助设计人员开发整合了 TI 产品的应用；如果您（个人，或如果是代表贵公司，则为贵公司）以任何方式下载、访问或使用了任何特定的 TI 资源，即表示贵方同意仅为该等目标，按照本通知的条款进行使用。

TI 所提供的 TI 资源，并未扩大或以其他方式修改 TI 对 TI 产品的公开适用的质保及质保免责声明；也未导致 TI 承担任何额外的义务或责任。TI 有权对其 TI 资源进行纠正、增强、改进和其他修改。

您理解并同意，在设计应用时应自行实施独立的分析、评价和判断，且应全权负责并确保应用的安全性，以及您的应用（包括应用中使用的 TI 产品）应符合所有适用的法律法规及其他相关要求。您就您的应用声明，您具备制订和实施下列保障措施所需的一切必要专业知识，能够 (1) 预见故障的危险后果，(2) 监视故障及其后果，以及 (3) 降低可能导致危险的故障几率并采取适当措施。您同意，在使用或分发包含 TI 产品的任何应用前，您将彻底测试该等应用和该等应用所用 TI 产品的功能。除特定 TI 资源的公开文档中明确列出的测试外，TI 未进行任何其他测试。

您只有在为开发包含该等 TI 资源所列 TI 产品的应用时，才被授权使用、复制和修改任何相关单项 TI 资源。但并未依据禁止反言原则或其他法律授予您任何 TI 知识产权的任何其他明示或默示的许可，也未授予您 TI 或第三方的任何技术或知识产权的许可，该等产权包括但不限于任何专利权、版权、屏蔽作品权或与使用 TI 产品或服务的任何整合、机器制作、流程相关的其他知识产权。涉及或参考了第三方产品或服务的信息不构成使用此类产品或服务的许可或与其相关的保证或认可。使用 TI 资源可能需要您向第三方获得对该等第三方专利或其他知识产权的许可。

TI 资源系“按原样”提供。TI 兹免除对 TI 资源及其使用作出所有其他明确或默认的保证或陈述，包括但不限于对准确性或完整性、产权保证、无复发故障保证，以及适销性、适合特定用途和不侵犯任何第三方知识产权的任何默认保证。

TI 不负责任何申索，包括但不限于因组合产品所致或与之有关的申索，也不为您辩护或赔偿，即使该等产品组合已列于 TI 资源或其他地方。对因 TI 资源或其使用引起或与之有关的任何实际的、直接的、特殊的、附带的、间接的、惩罚性的、偶发的、从属或惩戒性损害赔偿，不管 TI 是否获悉可能会产生上述损害赔偿，TI 概不负责。

您同意向 TI 及其代表全额赔偿因您不遵守本通知条款和条件而引起的任何损害、费用、损失和/或责任。

本通知适用于 TI 资源。另有其他条款适用于某些类型的材料、TI 产品和服务的使用和采购。这些条款包括但不限于适用于 TI 的半导体产品 (<http://www.ti.com/sc/docs/stdterms.htm>)、[评估模块](http://www.ti.com/sc/docs/sampters.htm)和样品 (<http://www.ti.com/sc/docs/sampters.htm>) 的标准条款。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122
Copyright © 2017 德州仪器半导体技术（上海）有限公司